Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Towards a Generalized Framework for Secure Time-Stamping

Masterarbeit

zur Erlangung des akademischen Grades
Master of Science (M. Sc.)

eingereicht von:	Keno Goertz
geboren am:	20.11.1999
geboren in:	Saarlouis
Gutachter/innen:	Prof. Dr. Matthias Weidlich Prof. Dr. Florian Tschorsch

eingereicht am: .... verteidigt am: ....

Abstract. Write your abstract here.

Contents

1 Introduction

1.1 Trusted time-stamping

The simplest approach to digital time-stamping relies on a trusted third party (TTP). If Alice wants to time-stamp a document and prove the document's existence at the time-stamp's time to Bob at some later time, she can ask a time-stamp authority (TSA) to cryptographically sign a secure hash of her document together with the current time. Bob accepts the TSA's signature as proof of the document's existence at the specified time. (cite Haber1991Timestamp)

This scheme requires complete trust of both Alice and Bob in the impartiality of the TSA. Bob needs to trust the TSA to keep its private key secure and to never produce time-stamps for the past (an attack which I will refer to as "backdating"). Alice needs to trust the availability of the time-stamping service provided by the TSA whenever she wants to time-stamp a document.

This trust in a single authority can be problematic in practice. Even if we could assume complete impartiality of the TSA with regard to Alice and Bob, what happens if the party responsible for running the TSA wants to time-stamp a document of their own? Clearly, to ensure impartiality, another TSA would need to be used. But now what if neither of our TSAs can be assumed to be impartial with regard to yet another party who wants to time-stamp a document? Manually keeping track of which TSA can be trusted under which circumstances quickly becomes impractical. The notion of distributed trust will simplify matters considerably.

1.2 Distributed trust

1.2.1 Publication and witnesses

Trusted time-stamping requires complete trust in the time-stamp authority. This does not mean, however, that the TSA is actually *trustworthy*. We can decrease the amount of trust that we need to put in any single party by distributing trust across multiple parties.

In the context of time-stamping, we can achieve this by requiring the TSA to *publish* its time-stamps to a large number of *witnesses*. The publication can be implemented in many different ways, which we will take a look at in more detail later. For now, the reader may imagine that the TSA publishes its time-stamps in a newspaper. The time-stamping company *Surety* actually employed this method of publication in practice. (Citation needed)

Witnesses keep a record of the time-stamps issued by the TSA. They do not accept time-stamps issued too far in the past. Staying with the example of time-stamps published in a newspaper, the newspaper archives of public libraries can act as witnesses. To prevent backdating attacks, a library only archives a newspaper which it receives on the printed date of publication.

When a client wants to verify the validity of a time-stamp, they can now ask a selection of witnesses for confirmation. Using our example of newspaper archives, a client visits a handful of library archives and confirms that the time-stamp in question is actually printed in the archived newspapers of that date. Clients only accept time-stamps for which they find a sufficient number of witnesses.

Using such a publication scheme, a malicious TSA can no longer carry out a backdating attack all by itself. Instead, it would require the active cooperation of a sufficiently large number of witnesses in order to convince a client of the validity of a backdated time-stamp. The client's trust is thus *distributed* over the TSA, the publication process and the witnesses.

1.2.2 Quantifying distributed trust

Let us now introduce a mathematical model for the publication scheme outlined in the previous section. Say the TSA publishes its time-stamps to N witnesses. It should be emphasized that a witness is required to keep a record of time-stamps. Going back to our example of time-stamps published in a newspaper, N does *not* correspond to the number of copies printed. Instead, N refers to the number of places that keep archives of the newspaper.

We assume that there exist a number E of malicious witnesses that collude together with the TSA in an attempt to backdate time-stamps.

Finally, a client consults a number n of witnesses to verify a time-stamp. The client only accepts the time-stamp if all n selected witnesses confirm its existence at the given time.

Let e be the number of maliciously colluding witnesses selected by the client. Evidently, a successful backdating attack occurs when the client selects only colluding witnesses, so when $e = n$.

Let us now further assume that the client selects its n witnesses from the total number of witnesses N completely at random. Our problem is now equivalent to the urn problem when "drawing without replacement". e thus follows the hypergeometric distribution. (cite Forbes2010Statistical pp. 117-119)

$$P(e = k) = \binom{E}{k}\binom{N-E}{n-k}\Big/\binom{N}{n} \tag{1}$$

The probability of a successful backdating attack is then given by the equation:

$$P(e = n) = \binom{E}{n}\Big/\binom{N}{n} \tag{2}$$

In practice, the selection of witnesses may not be truly random. Sticking to our example of newspaper archives, a client will likely prefer libraries which are geographically close to them. A network protocol for distributed trust may also favor witnesses with small round-trip times in order to increase performance.

An attacker may be able to leverage this by placing colluding witnesses at favorable locations. We can model this by introducing a weight parameter ω, where a malicious witness is ω times more likely to be selected than an honest witness. e then follows Fisher's noncentral hypergeomtric distribution. (cite Fog2008Sampling)

$$e_{\text{min}} = \max(0, n + E - N) \tag{3}$$

$$e_{\text{max}} = \min(n, E) \tag{4}$$

$$P(e = k) = \binom{E}{k}\binom{N-E}{n-k}\omega^{k} \Bigg/ \sum_{k'=e_{\text{min}}}^{e_{\text{max}}} \binom{E}{k'}\binom{N-E}{n-k'}\omega^{k'} \tag{5}$$

With the probability of a successful backdating attack being:

$$P(e = n) = \binom{E}{n}\omega^{n} \Bigg/ \sum_{k'=e_{\text{min}}}^{e_{\text{max}}} \binom{E}{k'}\binom{N-E}{n-k'}\omega^{k'} \tag{6}$$

Note that these equations are equivalent to the hypergeomtric distribution when $\omega = 1$. This is the optimal case, limiting the probability of a successful backdating attack as much as possible.

ω approaches infinity if the attacker can ensure that the client will only select malicious witnesses. In this case, the probability of a successful backdating attack approaches 1.

$$\lim_{\omega \to \infty} P(e = n) = 1 \tag{7}$$

This is, of course, the worst possible case for security.

TODO: Add lots of graphs for the probability distributions in this section.

TODO: The other side of trust is that Alice needs to trust service availability. Can be solved by employing multiple TSAs. Quickly explain this.

References

Appendix

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den March 18, 2025 ....